# A Constrained DMPs Framework for Robot Skills Learning and Generalization from Human Demonstrations

Zhenyu Lu, Ning Wang, *Member, IEEE,* and Chenguang Yang, *Senior Member, IEEE*

***Abstract*—Dynamical movement primitives (DMPs) model is a useful tool for efficiently robotic learning manipulation skills from human demonstrations and then generalizing these skills to fulfill new tasks. It is improved and applied for the cases with multiple constraints such as having obstacles or relative distance limitation for multi-agent formation. However, the improved DMPs should change additional terms according to the specified constraints of different tasks. In this paper, we will propose a novel DMPs framework facing the constrained conditions for robotic skills generalization. First, we conclude the common characteristics of previous modified DMPs with constraints and propose a general DMPs framework with various classified constraints. Inspired by barrier Lyapunov functions (BLFs), an additional acceleration term of the general model is deduced to compensate tracking errors between the real and desired trajectories with constraints. Furthermore, we prove convergence of the generated path and makes a discussion about advantages of the proposed method compared with existing literature. Finally, we instantiate the novel framework through three experiments: obstacle avoidance in the static and dynamic environment and human-like cooperative manipulation, to certify its effectiveness.**

***Index Terms*—Dynamic movement primitives (DMPs), robot learning, skills generalization, barrier Lyapunov functions(BLFs)**

## I. Introduction

Learning from demonstration (LfD), inspired by neuroscience, is an effective way for robotics learning manipulation skills such as opening doors or grasping cups from human's natural actions [1]. Since 1980s, plenty of methods are proposed, e.g. Calinon, Billard and Khansari-Zadeh et al. used Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) to regenerate motions [2]-[4]. Ng and Russell proposed the inverse reinforcement learning (IRL) method to build up an unknown reward function based on the observed trajectories to characterize solutions [5]. Ijspeert et al. proposed dynamic movement primitives (DMPs) model that produces a trajectory similar to the demonstrations to solve the path planning without building kinematic models [6], [7]. Since then, DMPs model was modified and improved widely for imitation and learning of human skills like reaching movements [8], object grasping [9] and clothes manipulation [10] etc.

Constraints in joint and task spaces [11] and caused by force interaction [12] are common topics for LfD. Some researchers improved DMPs model and combined neural networks control with DMPs [13]-[14] to solve the tasks with special constraints such as obstacle avoidance [15]-[19], bimanual operation [20]-[24] and interaction with external objects [27], the majority of which added a coupling term based on the basic functions.

For obstacle avoidance, Khansari-Zadeh et al. and Park et al. took repulsive potential fields as coupling terms into DMPs for obstacle avoidance [15], [16]. Hoffmann et al. motivated by biological data and human behaviors and modified the DMPs model by adding an acceleration term to avoid moving obstacle [17]. But, Pairet et al. pointed out that the additional coupling term will bring some limitations: dead-zone compromising the method's reliability, without a strategy to guide to a preferred route to circumnavigate an obstacle, performance decrease facing non-point obstacles, time-consuming and prone to measurement noise. Therefore, they proposed a hierarchical framework that combines the versatility of DMPs and strengths of learning techniques [18]. As statements mentioned in [16], these proposed approaches enable a robot to avoid conflict with an obstacle by predefining policies. But, a non-expert cannot teach a robot his/her special skills of avoiding a collision with different objects by demonstrations.

Cooperative DMPs for bimanual robot manipulation [20]-[24] and multi-robot formation [25] has another kind of constraint, which brings strong space constraints to all the related roles. Umlauft et al. [20] inspired by DMPs interacting with external perturbations, added an error item to increase adaptability under external disturbances. Kulvicius et al. studied interactive DMPs and added an accelerating predictive reaction for the coupling agents, which is similar to the term for obstacle avoidance [22].

This work was partially supported by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S001913 and the Key Research and Development Project of Zhejiang Province under Grant 2021C04017. (*Corresponding author*: *Chenguang Yang*)

The authors are with the Bristol Robotics Laboratory, University of the West of England, BS16 1QY, United Kingdom, (E-mail: luzhenyurobot@gmail.com; katie.wang@brl. ac.uk; cyang@ieee.org).

Gams et al. argued that not only at acceleration level but also the velocity should be added a term for smoother interactions in DMPs, which can be seen as the combination of above two papers [23]. They also used iterative learning control method to learn the coupling term to modify the original trajectory based on the force feedback generated during executions of the tasks. Lee et al. studied with purpose of both obstacle avoidance and cooperative manipulation by DMPs for aerial robots formation, and two terms were formulated based on regular DMPs to solve two problems separately [25]. Lastly, some researchers studied special constraints such as curved surface [26] to modify DMPs by adding an accelerating coupling term for realizing trajectory planning and force control [27].

Seen from the presented work, it is not hard to find that the coupling terms are specially designed with different constraints and purposes for the varying tasks and operational demands. In fact, these terms have two implicit usages: path replanning and generalization. As there are many researches has proposed path replanning methods for various scenes, we hope to propose a general DMPs-based skill generalization method based on the path replanning methods to enable the original path to fit tasks with constraints, especially, with several dispersed constraints. Some researchers have made a few experimental studies, e.g. in [28], Krug et al. combined the model predictive control (MPC) method and DMPs to calculate possibilities of obstacle avoidance for online optimization of trajectory. Furthermore, Khansari-Zadeh, and Billard proposed an idea of safety margin around obstacle via a safety factor for obstacle avoidance [15]. Compared with other researches, these two papers attempted to provide a general DMPs method to more cases with constraints. With this inspiration, we propose a novel DMPs framework that combines the control concept of Barrier Lyaponov Functions (BLFs) with regular DMPs model. The idea integrating control and learning methods is similar to our previous work [29]-[31], [38] but has a difference that is by adding an acceleration term calculated by constrained paths and safety margins, the control method will be embedded in the DMPs model and modify the trajectories timely. The main contributions are listed as follows:

1) Proposing a new abstracted DMPs model with classified constraints based on the summary of plenty of improved DMPs cases with constraints, covering obstacle avoidance, cooperative manipulation, and curved surface movement.
2) Proposing a new method combining path planning (DMPs) and control concept (BLFs) to solve constrained trajectory planning based on the new model. A technical discussion is taken to compare with methods presented in the previous papers[16]-[18]. Convergence of the generalized trajectory is proved.
3) Three experiments are taken to prove the effectiveness of the proposed method for non-point obstacle avoidance in static and dynamic environment and cooperative dual-arm robot operation. The results show that most of limitations of the original DMPs presented by Pairet et al [18] will be addressed.

The rest of the paper is organized as follows. Section 2 starts from basic knowledge of DMPs and proposes a new framework based on summary of modified DMPs with special constraints. The improved constrained DMPs is presented in Section 3. In Section 4, we make an analysis about advantages of the method comparing with existing literatures and present the procedures for skill learning and generalization in actual. Section 5 shows three DMPs-based experiments and compares the results with original DMPs. Section 6 concludes the paper finally.

## II. Dynamic Movement Primitives and Constraints

This section contains two parts. The first one introduces the basic components for DMPs function proposed by Ijspeert et al. [6]-[7], and the second part presents a general DMPs model with classified constraints, which is the research foundation for the next section.

### A. Dynamic movement primitives

The DMPs model is firstly proposed in [6] as

$$\begin{cases} \tau \dot{v} = \alpha_z \left( \beta_z \left( g - x \right) - v \right) + f\left( s \right) \\ \tau \dot{x} = v \end{cases}, \tag{1}$$

TABLE I
TYPICAL IMPROVED DMPS WITH CONSTRAINTS

| Ref. No. | Constraints' expressions | Expressions of improved DMPs |
|---|---|---|
| [17] | **Obstacle constraint**: $p(x,v) = \gamma R v \phi exp(-\beta\phi)$ , $R$ is a rotation matrix with axis of $r = (o-x) \times v$ and $o$ is the position of the obstacle, and $\phi = \cos^{-1}\left( \frac{(o-x)^T v}{\lvert o-x \rvert \cdot \lvert v \rvert} \right)$ | $\tau \dot{v} = k\left(g - x\right) - d\dot{x} - k\left(g - x_0\right)s + kf\left(s\right) + p(x,v)$ |
| [20] | **Cooperative distance constraint**: $c_i(x) = -\sum_{j \in N_i} \delta_{ij} \frac{x_i - x_j}{\left\| x_i - x_j \right\|} \left( \left\| x_i - x_j \right\| - d_{ij} \right)$ $d_{ij}$ is the desired distance between the agent $i$ and agent $j$; | $\tau \dot{v}_i = \alpha_z \left( \beta_z \left( g_i - x_i \right) - v_i \right) + f\left( s_i \right)$ $\tau \dot{x}_i = v_i + k c_i(x)$ $\tau \dot{s}_i = -\frac{\gamma_i s_i}{1 + \eta c_i^2(x)}$ |
| [23] | **Obstacle and internal force constraint**: $C_{i,j} = cF_{i,j} l_{fi}$ , $F_{i,j} = -F_{j,i} = -k(d_d - d_a)$ represents the internal force between two agents, and $d_d$ represents the desired distance between two effectors and $d_a$ is the actual distance; | $\tau \dot{v}_i = k\left( d\left( g - x_i \right) - \dot{x}_i \right) + f\left( s \right) + c_2 \dot{C}_{i,j}$ $\tau \dot{x}_i = v_i + C_{i,j}$ $C_{i,j} = cF_{i,j}(t)$ |
| [27] | **Curved surface constraint:** $\Delta a$ is added as the force coupling term related to the contact force error. | $\tau \dot{v} = k\left( d\left( g - x \right) - \dot{x} \right) + \alpha f\left( s, w \right) + \Delta a$ $\tau \dot{x} = v$ |

where $f(s)=\theta^T\Psi(s)$ is a linear combination of nonlinear radial basis functions, and $\theta=[w_1,w_2,...,w_n]^T$, $\Psi(s)=[\psi_1,\psi_2,...,\psi_n]^T$

$$\psi_i=\frac{\varphi_i(s)s}{\sum_{i=1}^{n}\varphi_i(s)},\varphi_i(s)=\exp(-h_i(s-c_i)^2)\cdot$$

where $c_i$ and $h_i>0$ are the centers and widths of radial basis functions respectively. The factors $\alpha_z,\beta_z>0$ are coefficients of the linear part in (1) and (1) has a unique attracting point at $x=g,v=0.\ \tau>0$ is a timing parameter adjusting speed before execution of movements. $s$ is a phase variable to achieve the dependency of function $f(s)$ out of time.

**Remark 1**: The expression of the forcing function is not unique, some researchers such as Wang et al. [32] and Wu et al. [33] proposed the DMP plus method by adding a bias term to each kernel and used the truncated kernels to achieve lower MSE in position deviation.

The dynamics of $s$ is expressed by a canonical system

$$\tau\dot{s}=-\gamma s,\qquad \gamma>0. \tag{2}$$

Eq. (2) has implicit relation with time that we can modify the converging time by the factor $\gamma$. When $s\to 0$, as $t\to\infty$, the value of $f(s)$ trends to 0 and $[x,v]$ reaches to the stability point $[g,0]$. The vector $\theta$ can be learned using supervised learning algorithms such as locally weighted regression (LWR) [32]-[34]. The purpose of calculating process is to minimize the error function:

$$\min\sum_{j=1}^{N}\left(f_j^{Tar}(s)-f_j(s)\right), \tag{3}$$

where $f_j(s)$ represents the item calculated by the *jth* trajectory in demonstration, and $f_j^{Tar}(s)$ is the target value of $f_j(s)$ as

$$f_j^{Tar}(s)=\tau\dot{v}-\alpha_z\left(\beta_z\left(g-x_j\right)-v_j\right). \tag{4}$$

*B. Generalization of modified DMPs and constraints*

As mentioned in Section I, eq. (1) is improved and proposed with several constraints , we list some typical ones in Table I. It is not hard to notice that both the basic expression of DMPs and additional terms differ from each other with various purposes for manipulations. We express them with a general function:

$$\begin{cases}\dot{v}=f_2(g,x,v,\tau)+k(s,\tau)+\beta_g u(s,\tau)+\Delta u\\ \dot{x}=f_1(x,\tau,F)+g_1(x,\tau)v\\ \dot{s}=\Phi(s,\tau)\end{cases}, \tag{5}$$

where $\Phi(s,\tau)$ is the canonical system like (2) and $k(s,\tau)$ and $u(s,\tau)$ containing $s$. The difference of $k(s,\tau)$ and $u(s,\tau)$ is $k(s,\tau)$ represents a linear term like $k(g-x_0)s$ in [17] and $u(s,\tau)$ is the forcing function like $f(s)$ in [17], [20] etc. $\beta_g$ is a constant factor and usually set as $1/\tau$. $f_2(g,x,v,\tau)$ is a general linear function of DMPs like $\alpha_z(\beta_z(g-x)-v)$ in (1). $\Delta u$ is an additional accelerating term caused by the constraints. All the DMPs functions in Table I can obviously abstracted into the formulation in (5). In the next section, a general method will be presented to deduce the exact expression of $\Delta u$.

Following Table I, we know that any constrained term has a referring point or trajectory like point $o$ in [17], or $x_j$ to $x_i$ in [20], and position errors like $o-x$ or $\|x_i-x_j\|-d_{ij}$ are used to calculate the additional term. As trajectory points are $x$, we set the referring point of $x$ as $x^c$, $x_i^c$ and $x_i$ are positions along the trajectories $x^c$ and $x$. Additionally, we are inspired by the concept of "safety limits" in [34] and "safety margin" in [15] and propose $h_i$ to describe the relations of $x_i^c$ and $x_i$. All the possible relations can be shown as:

$$\underline{h}_i\le x_i^c-x_i\le\bar{h}_i \text{ or } x_i^c-x_i\ge\bar{h}_i \text{ or/and } x_i^c-x_i\le\underline{h}_i. \tag{6}$$

where $\bar{h}_i$ and $\underline{h}_i$ are upper and lower boundary functions, and $\bar{h}_i\ge\underline{h}_i$. $x_i^c$ and $h_i$ (including $\bar{h}_i$ and $\underline{h}_i$) depends on operator's decisions or objective conditions learned on-time or by multi - disciplines in [16] and [18]. They also can be generated by local constraints and operator prefer routes. Different from intelligent learning methods of path planning, the proposed DMPs model has two following advantages:

1. Ensuring stability of trajectory. No matter the two-order expression of DMPs or the integrated BLFs method are proved to be stable and converged, but some intelligent methods are not.
2. DMPs can be integrated with intelligent learning methods and control theory to improve the learning efficiency and autonomy such as terms $x^c$ and $h_i$ can be manually set and learned by intelligent methods or calculated based on the analytic geometrical relations of interactive objects. After getting $x_i^c$ and $h_i$, the improved DMPs model will enact to ensure convergence of the generalized trajectory satisfying inequalities in (6).

Furthermore, we divide (6) into two conditions, which both contain two cases.

**Condition 1:** Asymmetric constraints

Case 1: $\underline{h}_i\le x_i^c-x_i\le\bar{h}_i$;

Case 2: $x_i^c-x_i\ge\bar{h}_i$ and/or $x_i^c-x_i\le\underline{h}_i$.

**Condition 2:** Symmetric constraints

Case 1: $\left|x_i^c-x_i\right|\ge\bar{h}_i$

Case 2: $\left|x_i^c-x_i\right|\le\underline{h}_i$

Some assumptions are proposed as follows:

**Assumption 1:** The constrained trajectory is stable and satisfies if $s\to 0$, then

$$\dot{v}^c = f_2\left(g, x^c, v^c, \tau\right) \text{and } \left(x^c, v^c\right) \to (g, 0). \quad (7)$$

**Assumption 2:** The constrained trajectories and velocities are continuous and differentiable within every skill segmentation range.

**Remark 2**: Assumption 1 is basis for trajectory convergence. According to [19], if $s \to 0$, the $s$-related term will decrease to 0 for normal DMPs function. While for (5), $\dot{v} \to f_2\left(g, x, v, \tau\right) + \Delta u$ which means that trajectory is influenced by error terms $z_1, z_2$. It is hoped that the additional constraints will not influence trajectory converging to the target, thus the generated referring trajectory satisfying Assumption 1 ensures that $x \to x^c$ and $v^c, v \to 0$ at the destination.

**Remark 3**: Assumption 2 is proposed for calculation. Actually, a long-term demonstration will be segmented into several parts. The constraints may be presented discretely aside the trajectory or even go across several segmented intervals. Two methods for generating continuous referring paths to connect the constraints is introduced in the next section, and Assumption 2 is the basis for the method.

## III. Constrained Dynamic Movement Primitives

### A. Constrained referring trajectory

As it is presented in Table 1, the referring variable $x^c$ maybe a point, a surface or a moving trajectory and its function is to guide $x$ to satisfy conditions in (6), though some researchers proposed constrained dynamic movement primitives (CDMPs) that does not build referring trajectory but use the transformed states instead of the original DMPs states to maintain the joint trajectories within the safety limits [34].

Creating referring trajectory points $x_i^c$ is the first step for the new DMPs method. Here, we consider a case containing some local separate constraints and build a trajectory reserving the majority of the original path points and using the constraints to avoid obstacles or pass narrow avenues. Then using the integral and continuous trajectory to guide robotic movements. Here, we provide two methods to achieve this purpose: interpolation and extended DMPs method.

Set intervals divided by constraints and skill segmentations are $[x_0, x_1], [x_2, x_3], ..., [x_{i-1}, x_i]$. The interpolation method such as Lagrange interpolation polynomial provides some internal points to enrich the space between adjacent intervals, ensuring the continuity of positions and velocities. The extended DMPs method uses the generalized sub-skills to modify the interval goals, which is achieved by two steps:

1) Calculating original DMPs within the interval $[x_{i-1}, x_i]$ and endpoint satisfying $g_i = x_i$;
2) Changing goal $g_i$ to the start of new period $x_{i+1}$, then the current interval is extended from $[x_{i-1}, x_i]$ to $[x_{i-1}, x_{i+1}]$.

The two methods are verified and compared in experiment 1. When the constraints come across with each other, a common solution satisfying all the constraints will be selected. Moreover, we design the boundary functions according to manipulation requirements. Following convergence proof in part C, trajectory will converge based on the condition of $z_1(s_i^{0+}) \in \left(\underline{h}_i, \bar{h}_i\right)$, where $s_i^{0+}$ represents $s_i$ in the $ith$ interval at the start time $0+$. Then the junction point for the $(i-1)$th and $i$ th intervals satisfies the condition of (6), such as for **case 1, condition 1**, we have

$$\left(x_{i-1}^c - x_{i-1}\right)\Big|_{s_{i-1}^{\infty}} = \left(x_i^c - x_i\right)\Big|_{s_i^{0+}} \in \left(\underline{h}_{i-1}, \bar{h}_{i-1}\right)\Big|_{s_{i-1}^{\infty}} \subseteq \left(\underline{h}_i, \bar{h}_i\right)\Big|_{s_i^{0+}}$$

to ensure continuity and avoid phase initially instability of path points.

### B. Constrained dynamic movement primitives

Barrier Lyapunov functions (BLFs) is used for analyzing stability of closed-loop system, and it restricts full-state to the constraints [29], [35]. In this paper, BLFs is adopted to calculate the additional item, enabling the trajectory calculated by new DMPs to satisfy constraints in (6).

Define two new variables $z_1 = k_z\left(x_i^c - x_i\right)$, $z_2 = d_z\left(\alpha_2 - v_i\right)$, $k_z, d_z > 0$ are constant, and $\alpha_2$ is a stabilizing function to be designed. Similar to asymmetric BLFs candidate in [36], we taking **case 1, condition 1** as an example and build a Lyapunov function as

$$V^c = \frac{1}{2}\left(q(z_1)\log\frac{\bar{h}_i^2}{\bar{h}_i^2 - z_1^2} + (1-q(z_1))\log\frac{\underline{h}_i^2}{\underline{h}_i^2 - z_1^2}\right) + \frac{1}{2}z_2^2. \quad (8)$$

where $q(\bullet) = 1, if\ \bullet > 0$ and $q(\bullet) = 0, if\ \bullet \le 0$. Then $\dot{V}^c$ is

$$\dot{V}^c = \left(\frac{q(z_1)}{\bar{h}_i^2 - z_1^2} + \frac{1-q(z_1)}{\underline{h}_i^2 - z_1^2}\right) z_1 k_z\left(\dot{x}_i^c - \dot{x}_i\right) + z_2 d_z\left(\dot{\alpha}_2 - \dot{v}_i\right). \quad (9)$$

For realizing the stability condition of $\dot{V}^c \le 0$, the terms $\alpha_2$, $u(s,\tau)$ and $k_n$ are taken as

$$\begin{cases} \alpha_2 = \dfrac{d_z\left(\dot{x}_i^c - f_1(x,\tau,F)\right) + z_1}{d_z g_1(x,\tau)} \\ u(s,\tau) = \dfrac{\dot{\alpha}_2 - f_2\left(g,x,v,\tau\right) - k\left(s,\tau\right) - \Delta u + k_n z_2}{\beta_g} \\ k_n = \dfrac{k_2}{d_z} + \dfrac{k_z g_1(x,\tau)^2}{4d_z^2}\left(\dfrac{q(z_1)}{\bar{h}_i^2 - z_1^2} + \dfrac{1-q(z_1)}{\underline{h}_i^2 - z_1^2}\right) \end{cases}, (10)$$

where $k_2 > 0$ is a positive number and the proof is presented in Part C. Following (4), $u^{Tar}$ without constraints is

$$u^{Tar} = \frac{\dot{v} - f_2\left(g,x,v,\tau\right) - k\left(s,\tau\right)}{\beta_g}. \quad (11)$$

It is desired that the additional constraints do not repeat the skill learning process (calculation of forcing functions), then the value error of $u(s,\tau) - u^{Tar}$ will decrease to zero, and $\Delta u$ is

$$\Delta u = \dot{\alpha}_2 + k_n z_2 - \dot{v}\,. \tag{12}$$

Taking (12) into (5), we get the modified DMPs satisfying constraints in **case 1, condition 1** as

$$\begin{cases} \dot{v} = f_2(g,x,v,\tau) + k(s,\tau) + \beta_g u(s,\tau) + \Delta u \\ \dot{x} = f_1(x,\tau,F) + g_1(x,\tau)v \\ \dot{s} = \Phi(s,\tau) \\ \Delta u = \dot{\alpha}_2 + k_n z_2 - \dot{v} \end{cases}. \tag{13}$$

**Remark 4:** For the regular DMPs model like (1), the terms in (13) are instantiated as $f_1(x,\tau,F)=0, g_1(x,\tau)=1/\tau$, $f_2(g,x,v,\tau) = \alpha_z\left(\beta_z(g-x)-v\right)/\tau, k(s,\tau)=0, \beta_g = 1/\tau$, then

$$\begin{cases} \tau\dot{v} = \alpha_z\left(\beta_z(g-x)-v\right) + f(s) + \tau\left(k_n z_2 + \dot{z}_2/d_z\right) \\ \tau\dot{x} = v \\ \tau\dot{s} = -\gamma s \end{cases}. \tag{14}$$

**Case 2:** Similar to (8) of case 1, we build the BLFs candidate :

$$\begin{aligned} V^c = {} & \frac{1-q(z_1)}{2}\left((1-q(\underline{h}_i))\log\frac{z_1^2}{z_1^2-\underline{h}_i^2} + (1-q(\bar{h}_i))\log\frac{\bar{h}_i^2}{\bar{h}_i^2-z_1^2}\right) \\ & + \frac{q(z_1)}{2}\left(q(\underline{h}_i)\log\frac{\underline{h}_i^2}{\underline{h}_i^2-z_1^2} + q(\bar{h}_i)\log\frac{z_1^2}{z_1^2-\bar{h}_i^2}\right) + \frac{1}{2}z_2^2 \end{aligned}, \tag{15}$$

and the factors satisfying $\dot{V}^c \le 0$ are calculated with different results in (10) except for the $u(s,\tau)$ as

$$\begin{cases} \alpha_2 = \dfrac{\dot{x}_i^c - f_1(x,\tau,F)}{g_1(x,\tau)} - \dfrac{K_2 + K_1 z_1^4}{d_z g_1(x,\tau)\left(K_2 - K_1 z_1^3\right)} \\ k_n = \dfrac{k_2}{d_z} + \dfrac{k_z g_1(x,\tau)^2}{4d_z^2}\left(K_1 + K_2\right) \\ K_1 = \dfrac{\left(1-q(z_1)\right)\left(1-q(\bar{h}_i)\right)}{\bar{h}_i^2 - z_1^2} + \dfrac{q(z_1)q(\underline{h}_i)}{\underline{h}_i^2 - z_1^2} \\ K_2 = \dfrac{\underline{h}_i^2\left(1-q(z_1)\right)\left(1-q(\underline{h}_i)\right)}{z_1^2 - \underline{h}_i^2} + \dfrac{\bar{h}_i^2 q(z_1) q(\bar{h}_i)}{z_1^2 - \bar{h}_i^2} \end{cases} \tag{16}$$

**Condition 2**: Lyapunov functions are built with a different formation to (8), and $V^c$ for cases 1 and 2 are defined as

TABLE II
SYMBOL CALCULATIONS FOR CONDITION 2

| Symbol | $\left\|x_i^c - x_i\right\| \ge \bar{h}_i$ | $\left\|x_i^c - x_i\right\| \le \underline{h}_i$ |
|---|---|---|
| $\alpha_2$ | $\dfrac{\dot{x}_i^c - f_1(x,\tau,F)}{g_1(x,\tau)} - \dfrac{1}{d_z g_1(x,\tau) z_1}$ | $\dfrac{\dot{x}_i^c - f_1(x,\tau,F)}{g_1(x,\tau)} + \dfrac{z_1}{d_z g_1(x,\tau)}$ |
| $u(s,\tau)$ | $\dfrac{\dot{\alpha}_2 - f_2(g,x,v,\tau) - k(s,\tau) - \Delta u + k_n z_2}{\beta_g}$ | |
| $k_n$ | $\dfrac{k_2}{d_z} + \dfrac{k_z\left(\bar{h}_i g_1(x,\tau)\right)^2}{4d_z^2\left(z_1^2 - \bar{h}_i^2\right)}$ | $\dfrac{k_2}{d_z} + \dfrac{k_z\left(g_1(x,\tau)\right)^2}{4d_z^2\left(\underline{h}_i^2 - z_1^2\right)}$ |

$$V^c = \frac{1}{2}\log\frac{z_1^2}{z_1^2 - \bar{h}_i^2} + \frac{1}{2}z_2^2 \text{ and } V^c = \frac{1}{2}\log\frac{\underline{h}_i^2}{\underline{h}_i^2 - z_1^2} + \frac{1}{2}z_2^2 \tag{17}$$

Factors $\alpha_2$, $u(s,\tau)$ and $k_n$ enable system stability condition $\dot{V}^c \le 0$ to be satisfied are shown in Table II.

**Remark 5:** Compared with (8), it is not hard to find that the expression of $u(s,\tau)$ does not change along with constraints, such that $\Delta u$ is calculated with a fixed expression for any cases, though the other two factors $\alpha_2$ and $k_n$ will vary with different conditions.

### *C. Convergence proof*

Some previous work integrated control methods with DMPs and proved convergence of the generated DMPs trajectory. In [19], Rai and Meier et al. built a Lyapunov stability criterion for DMPs with a coupling term to prove the condition that if $s \to 0$ or $t \to \infty$, then $x \to g$, which means that the convergence of trajectory is changed by the additional term $C^t$. Similarly, for $\Delta u$ in (13), we synthesize a Lyapunov function as

$$V = V^1 + V^c = \frac{1}{2}(g-x)^T K(g-x) + \frac{1}{2}v^T v + V^c\left(x^c, x\right), \tag{18}$$

where $K = \alpha_z\beta_z > 0$, $V^1$ represents the two former terms of $V$. Taking case 1, condition 1 as an example, to prove $\dot{V} < 0$, we first proves $\dot{V}^c \le 0$. Taking (5) into (9), we have

$$\begin{aligned} \dot{V}^c = {} & z_1 k_z\left(\frac{q(z_1)}{\bar{h}_i^2 - z_1^2} + \frac{1-q(z_1)}{\underline{h}_i^2 - z_1^2}\right)\left(\dot{x}_i^c - f_1(x,\tau,F) - g_1(x,\tau)\left(\alpha_2 - \frac{z_2}{d_z}\right)\right) \\ & + d_z z_2\left(\dot{\alpha}_2 - f_2(g,x,v,\tau) - k(s,\tau) - \beta_g u(s,\tau) - \Delta u\right) \\ = {} & z_1 k_z\left(\frac{q(z_1)}{\bar{h}_i^2 - z_1^2} + \frac{1-q(z_1)}{\underline{h}_i^2 - z_1^2}\right)\left(\dot{x}_i^c - f_1(x,\tau,F) - g_1(x,\tau)\alpha_2 + \frac{z_1}{d_z}\right) \\ & + d_z z_2\left(\dot{\alpha}_2 - f_2(g,x,v,\tau) - k(s,\tau) - \beta_g u(s,\tau) - \Delta u + \right. \\ & \left.\left(\frac{k_2}{d_z} + \frac{k_z g_1(x,\tau)^2}{4d_z^2}\left(\frac{q(z_1)}{\bar{h}_i^2 - z_1^2} + \frac{1-q(z_1)}{\underline{h}_i^2 - z_1^2}\right)\right) z_2\right) - \\ & \frac{k_z}{d_z}\left(\frac{q(z_1)}{\bar{h}_i^2 - z_1^2} + \frac{1-q(z_1)}{\underline{h}_i^2 - z_1^2}\right)\left(z_1 - \frac{g_1(x,\tau)z_2}{2}\right)^2 - z_2^T k_2 z_2 \end{aligned}. \tag{19}$$

Using the symbols in (10), we have

$$\dot{V}^c = -\frac{k_z}{d_z}\left(\frac{q(z_1)}{\bar{h}_i^2 - z_1^2} + \frac{1-q(z_1)}{\underline{h}_i^2 - z_1^2}\right)\left(z_1 - \frac{g_1(x,\tau)z_2}{2}\right)^2 - z_2^T k_2 z_2 \,. \tag{20}$$

Because $k_z, d_z, k_2 > 0$, $\bar{h}_i^2 - z_1^2 > 0$, and $\underline{h}_i^2 - z_1^2 > 0$, then if $z_1(s_i^{0+}) \in \left(\underline{h}_i, \bar{h}_i\right)$, where $s_i^{0+}$ is the phase variable equals to the time $t = 0^+$ for the $k$th interval, then the inequality $\dot{V}^c \le 0$ holds.

According to remark 2, we know that when $s \to 0$, we have $\dot{V}^c\left(x^c, x\right) \le 0$ and $z_1, z_2, \dot{z}_2 \to 0$, then (13) will degenerate to

(1). Following [19], $V^1 < 0$ will be proved under the condition that if $s \to 0$, then $f(s) \to 0$. Then combining assumption 1, when $s \to 0$, we have $\tau\dot{v} = \alpha_z(\beta_z(g-x)-v)$. Setting $K = \alpha_z\beta_z$, we can get

$$\begin{aligned}\dot{V}^1 &= -\dot{x}^T K(g-x) + v^T\dot{v} \\ &= -\frac{v^T}{\tau}K(g-x) + \frac{\alpha_z}{\tau}v^T(\beta_z(g-x)-v). \\ &= -\frac{\alpha_z}{\tau}v^T v < 0\end{aligned} \quad (21)$$

It is easy to know that $\dot{V} \le 0$ is achieved. The proof reveals that the additional term does not influence the system stability.

## IV. Discussion

In this section, we will make a discussion about if the method can address most of the problems stated by Pairet et al in [18]. Here, we classify the problems into following three categories and most of them are proved through experiments in Section V.

**Dead-zone & preferred route**: In [18], the authors stated that for the point obstacle avoidance method like [17], the analytical term $\dot{\theta} = \gamma\theta exp(-\beta\theta)$ ($\beta$ is a constant and $\theta$ is an variable) has a dead zone around 0, and the system will become less reactive as the heading towards the obstacle narrows, thus compromising the method's reliability. But for the proposed method, we can select multi-piece trajectories or make a tiny modification of the planned path around the dead zone as new references, and the preferred route also can be designed flexibly following the operator's preference.

**Moving non-point obstacles**: Some papers used DMPs model for the case with moving obstacles, such as Hoffmann et al. studied the robustness of dynamic systems against perturbations for obstacle avoidance and extended it to the cases with multiple obstacles and moving obstacles [17]. Park et al. designed a special dynamic potential function for the moving obstacles [16]. But, these methods don't suit non-point obstacle cases. As some existed researches addressed the object detection and real-time trajectory navigation based on autoregressive model [37], Kalman filter or tracking method [39] to solve the problem for non-demonstration cases, we can select a suitable method as prepositive treatment of our proposed DMPs framework, but it is not the main work in this paper.

By using detection method , we assume for any position $x_i$, robot can perceive edges and forecast dynamic positions of the obstacle within the next few steps as $X^o = \left[x_k^o, x_{k+1}^o, ..., x_{k+N}^o\right]^T$ and $x_k^o$ is a data set filled with detected obstacle positions. Then path point $x_k^c$ and constraints are generated by setting a 'safety margin' [15] toward the obstacle. The advantages of combining path prospective method and proposed DMPs model are:

1) Without the need of prior knowledge for all the objects in the whole map and predesigned trajectories e.g. [16]. The robot can select a planned path to avoid conflicts and return to the original DMPs route once facing obstacles.
2) Useful for multi-segment convex obstacle avoidance. As the presentations in the experiment of Section V.B that the referring path connects several local constraints to form a continuous long trajectory, and most of the constraints can be designed with different conditions change with dynamic environment timely and flexibly.

**Time-consuming & noise**: Actually, the calculation in (13) is more complex than the methods in [16], [17]. But, the proposed method just modifies trajectory points along part of the previous path, which can avoid extra calculation for the whole trajectory. For the influence of measuring noise, it is hard to be diminished but we can estimate the boundaries of noise varying range and add them for the design of $h_i$. For example, in experiment 3, using physical constraints (e. g. distance of joint links and end effectors), the measuring noises and errors will be filtered and amended. Though the proposed method solves most problems with universal equations for various cases, similar to most of the path replanning methods, it depends on prediction accuracy of the environmental parameters and referring trajectories that can be calculated timely.

Another issue for discussion is calculation procedures (Fig.1) of the improved DMPs method in calculation.

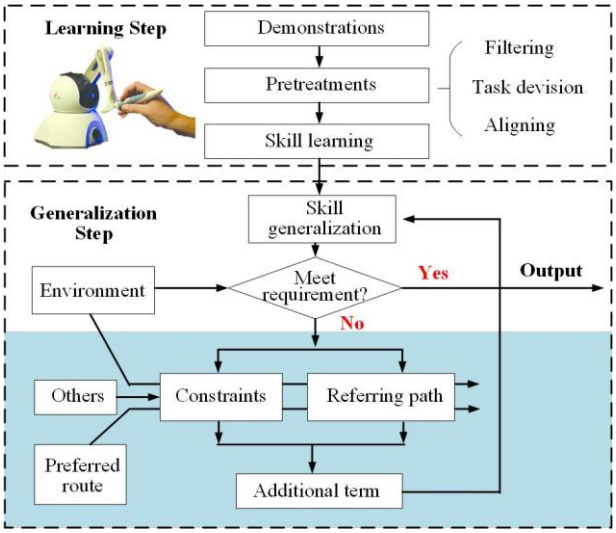



Fig. 1. Calculation procedures of the constrained DMPs framework

Similar to normal skill learning process based on DMPs, there are two steps from demonstrations to applications in new cases. After pretreatments such as data filtering, task division and data aligning etc., the forcing function will be learned and then the skill (trajectory) will be generalized based on the given new start and end point and timing factor. But, due to the new cases occurred in the environment, the primitive generated path may not satisfy the operational demands. Following the changes of the environment, preferred routes of the operators and some physical limitations will be used to generate the referring path (Section III.A) with classified constraints (Section II.B). Using eqs.(10) and (12), the additional term $\Delta u$ will be calculated and added to DMPs functions and the new trajectory will be generalized. The term $\Delta u$ does not influence the skill learning results and easy to be embedded to the original DMPs function.

The chosen parameters such as $\tau, \alpha_z$ and $\beta_z$ are the same as in (3) and (4). In the next section, we will follow the diagram to explore the potential applications of the method for different tasks with various conditions.

## V. Experiments

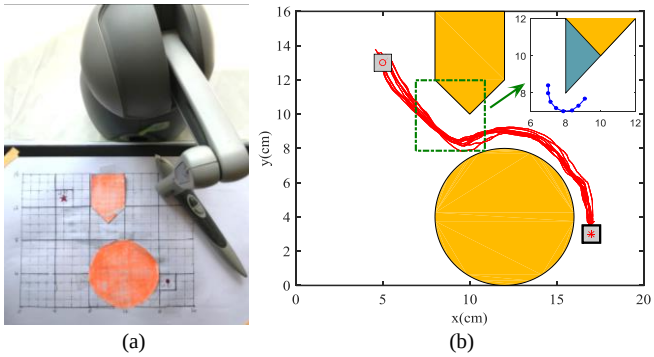
We take three experiments for various kinds of constraints caused by human preferences, shapes of extra obstacles, and physics limitations. The first experiment is about static non-point obstacle avoidance to certify special zone avoidance and preferred route selection. The second is about dynamic convex obstacle avoidance to solve the second kind of problems in the previous section. The last is about dual-arm robot cooperative operation with distance constraints of end effectors and joint links. Influence of measuring noise and errors are weakened by using physical constraints. The former two experiments use Phantom Omni joystick to record demonstrator's operational trajectories and the third uses Kinect to track skeleton results.

The experiments select DMPs expression in (14) to generate trajectories. The common parameters are $k_z = 150, d_z = 25$, $\alpha_z = 25, \beta_z = \alpha_z/4, \gamma = 0.7, k_2 = 10$, and others are set separately in each experiment.

### A. Static obstacle avoidance

The experimental setup is shown in Fig. 2. The platform is built on a board with a 2D map which is a square of $20 \times 16$ cm and the startpoint locates at [17, 3] and endpoint at [5, 13]. The values are recorded and presented in the virtual environment of MATLAB in Fig.2 (b). Here we set predesigned preferred trajectory points $x_i^c \in R^{2\times 1}$ (blue line with dots in the zoomed subfigure) to avoid confliction with additional block (blue triangle area). Furthermore, we set a "safety margin" $h_i = 0.5$ to limit the trajectory ranges that satisfies:

$$\left\| x_i^c - x_i \right\|_2 \le h_i, x_i \in R^{2\times 1}. \quad (22)$$




(a) (b)

Fig. 2. Experimental setups and conditions for static obstacle avoidance (**a**) Experiment equipment (**b**) Simulation environment equivalent to the physical: orange blocks are original obstacles and the triangular blue block is additional obstacle and the blue arc is the prospected trajectory with a safe distance to the new obstacle to protect its corner. Human demonstrated motions are colored in red and have no conflict with original obstacles.

Fig. 3 shows the generated constrained trajectories and simulation results. Using BP-AR-HMM method mentioned in [43], demonstrations are segmented and generalized into two sub-skills (red and blue lines in Fig.3 (a)), each of which is reshaped with 100 points. Due to the new obstacle, the robot following the path will encounter a collision. Furthermore, the constraints only affect the second sub-skill points, judging by the coordinate ranges in the X-axis. Therefore, we redivide the ranges of the second sub-skill into three regions (two black and a red line), and generate the whole-range potential trajectory points by the two methods developed in section IV.B

Fig.3 (b) shows a referring path with constrained path point generated by cubic spline interpolation that reserves part of the results of original DMPs. It connects original path point sets and constrained path with optimized interpolation points. While the trajectories generated by the extended DMPs model, shown in Fig.3(c), come cross the original obstacles and are not suitable for the task in this paper. The main reason is environmental factors (shape of obstacles and conflict etc.) are not considered in the original DMPs function. Thus, we select Fig.3 (b) as the referring path in this paper. But, it does not mean that which of the two methods is better, but depends on operational demands for different tasks. Usually, for a short distance point-to-point connection, the interpolation method is easier to be used and it will not change the previously planned path, while the DMPs-based method changes the distinction of the trajectory segment, which can be generalized in the space and time scales. While, the obstacles and other limitations should be considered during the design process for both cases.

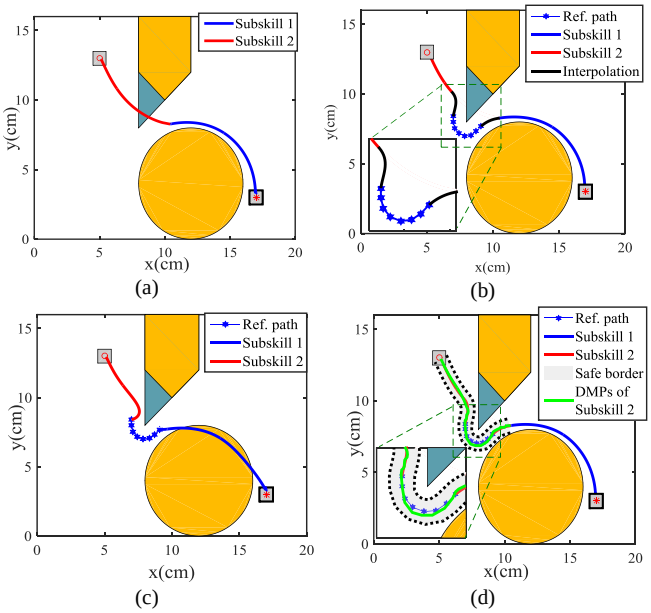



Fig. 3. Experimental results of static obstacle avoidance **(a)** Trajectories of original DMPs model, containing two segmentations: blue line for the first and red line for the second; **(b)** Constrained path generated by interpolation method: thin blue line with dot is the constrained path, blue line is trajectory of subskill 1 and red line is the reserved original subskill 2, and black line the interpolation trajectory; **(c)** Constrained path generated by extended DMPs; thin blue line with dot is the constrained path, and the blue and red line are the extend DMPs trajectories to connect constrained path; **(d)** Final generated trajectories based on the constrained route in subfigure (b), compared with (b), the additional area and lines are: green line is the generated 2nd DMPs subskill and the gray area is the safe margin to protect collision with obstacles.

Finally, Fig.3(d) shows the final path generated by modified DMPs. Under the conditions in (22), it researches to the largest

distance to the constrained path, but still keeps within a certain distance to $x_2^c$ and converges to constrained path quickly at the end of arch points.

**Remark 6:** Here, we make a further analysis about the selection of parameters. Actually, the parameters $\alpha_z$ and $\beta_z$ are selected as $\beta_z = \alpha_z/4$ to guarantee that the globally stable system in (1) and (14) are critically damped, allowing $x$ to monotonically converge to $g$ [32]. Factors $k_z$ and $d_z$ relate to the position and velocity errors, and a similar rate to $x$ and $v$ is selected as $k_z = d_z^2/4$. Here, we select an approximate solution as $k_z = 150$ $d_z = 25$. The simulation shows that $k_2$ realizes system stability and influences dynamic performance of the generate trajectory, A larger $k_2$ will achieve a larger tracking error of $x - x^c$ for this experiment, but it is not a conclusive result for all cases.

### *B. Moving convex obstacle avoidance*

The purpose of this experiment is to certify real-time obstacle avoidance with the consideration of the shape of obstacles. The simulation condition and initial demonstrations are as the same as experiment one and there are two obstacles colored in yellow and blue, as Fig. 4 shows. Considering the DMPs path does not have direct relations with time, we set the dynamic obstacle colored in orange to move with a speed of [0.005,0.006] every step and the movements can be observed by the robot. By using the method mentioned in Section VI.C and DMPs model of (1), we firstly estimate $x$ within the next ten steps and let robot to detect obstacles within a circular area. The obstacle edges in the area will be calculated the distance to $x$, and only the one with minimum distance will be selected to design point $x_i^c \in R^{2\times1}$. To avoid conflict caused by estimated error, we set constraints as $\left\| x_i^c - x_i \right\|_2 \le 0.8,\ x_i \in R^{2\times1}$. If there is no obstacle, the path points will be calculated by (1) as before.

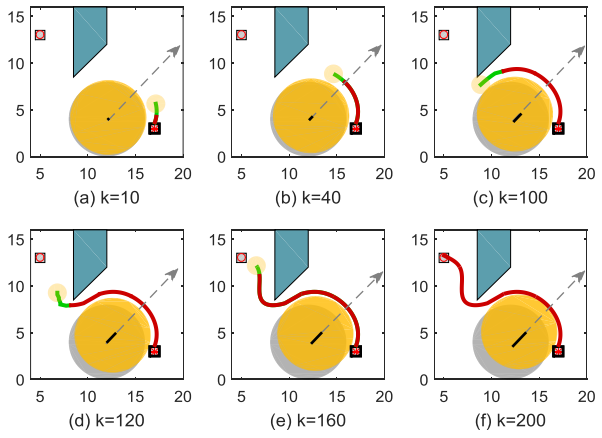



Fig. 4. Experimental results for dynamic obstacle avoidance: the blue block is a static obstacle and the orange block is a dynamic obstacle with a slow speed of [0.005,0.006] every step, and the gray arrow represents the moving direction and the black line segment is the moved distance of current state. To present the moving state of the mobile obstacle, the gray round area represents the original location. As the end point design the constrained path and margins timely, the green line represents the constrained path within next ten steps with the yellow detection area.

The simulation takes 200 iterative steps and we select some internal simulation results to present in fig.3. It can be seen that the trajectories avoid collision with all the obstacle edge points and reach to the distinction. Because $x$ are determined by the obstacles and the original DMPs model, the generated trajectory is closer to the obstacle boundary than in experiment one.

**Remark 7:** The constraints of experiment 1 and 2 that limit the generated path within a range around the referring path, can be realized by multiple unilateral constraints, which is similar to the propositions in [15]. The constrained referring trajectory $x_i^c$ can be seen as the desired path. Without building safety margin for the generated path, we can use modified DMP methods such as DMP plus [32], [33] to fit the constrained path points, which has smaller lower MSE in position deviation compared with ordinary DMPs. The general expression of (5) is compatible to different DMPs regression methods. By setting $h_i$ as the safety margin, the generated path has larger freedom to move within the safety area and easy to acquire smoother results. Meanwhile, the perfect dynamic performance of DMP plus method will help to provide a wider varying space than the normal DMPs.

### *C. Bimanual cooperative manipulation*

DMPs are widely used for multi-joint humanoid skill learning such as object-lifting, and cutting food [40]-[44]. Most of the papers only learned and generalized joint information [40], [41], [43], while for cooperative bimanual manipulation with a strict distance limitation of robot end effectors, e.g. holding a box or grasping a bar, only training data for joint information is not enough. Thus, cooperative DMPs [20]-[24] usually studied the trajectories of robot ends in the Cartesian coordinate. In this experiment, we combine these two cases and consider the constraints both in the joint distance and relative distance of robot ends. First, we build a data set about positions of the shoulders, elbows and wrists (for simplification, we use the wrists data instead of hands) of demonstrations measured by Kinect. there are two major problems of the measured data:

1) Positions of the wrists exist large numbers of errors due to the occlusions by the chair back and opaque object in Fig.5. Then the trained results of DMPs by the raw data are prone to exist errors;
2) Compared with the wrists and the elbows, the data of the shoulders is more stable and creditable. As it is shown in Fig. 6, we calculate internal distances of the wrists, elbows and shoulders to middle values of 10 times demonstrations. It can be seen that the data width of the wrists is about as twice as the shoulders.

Because the cooperative manipulation of robot ends endures the constraint of constant relative distance, by defining $x_j^w, x_j^e$ $x_j^s \in R^{3\times1}, j = l, r$ as positions of the wrists, elbows and shoulders, we proposed the following constraints for robot end effectors:

$$wrist: \begin{cases} \left\| x_j^{dw} - x_j^w \right\|_2 \le h_j^w \\ x_j^{dw} = x_j^w + d_j^w \end{cases}, j = l, r, \qquad (23)$$

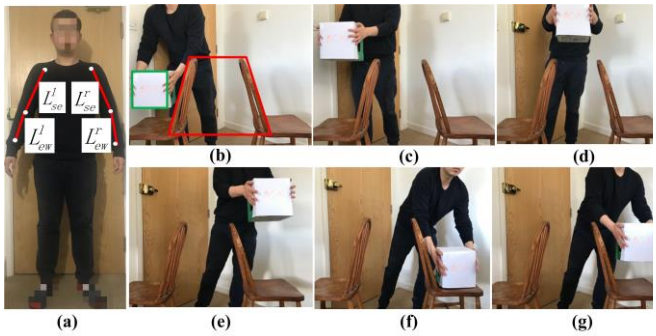



Fig. 5. Bimanual holding and replacing an opaque box (a) Coordinate of human body and related links length (b)~(f) Moving steps of the object

where $d_j^w \in R^{3\times1}$ represents the desired relative distance and $\left|d_j^w\right|$ is a constant , and $h_j^w$ is an acceptable distance calculation errors caused by object deformation or measuring errors. Define $L_{se}^j$, $L_{ew}^j$ are the vectors between the shoulders and the elbows, and the elbows and the wrists (Fig.5(a)) , the constraints of other linked joints are:

$$shoulder:\begin{cases}\left\|x_j^{ds}-x_j^s\right\|_2 \le h_j^s \\ x_j^{ds}=L_{se}^j+x_j^e\end{cases}, elbow:\begin{cases}\left\|x_j^{de}-x_j^e\right\|_2 \le h_j^e \\ x_j^{de}=L_{ew}^j+x_j^w\end{cases}, \ j=l,r \quad . \tag{24}$$

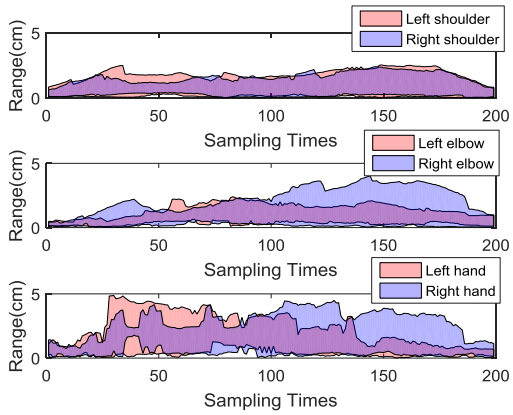



Fig. 6. Spreading distances of the wrist, elbow and shoulder. In each subfigure, red area represents distance range of the left hand, and blue area is the results of the right hand, and the purple area presents the intersection area.

Due to the mentioned relative distance constraints (eq.(23)) and distance constraints (eq.(24)) are contradicted: every joint is affected by relative constraints, but the position error rates of the end effectors are higher than the other joints, we propose the following calculation diagram:

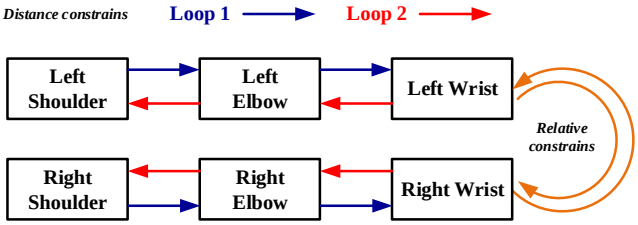



Fig. 7. Calculation diagram for cooperative DMPs that contains two loops of distances constraints and one relative constraints

The distance constraints are utilized twice: in loop 1, we use the shoulder and elbow data to correct the wrist data with errors. Then, the relative positions of the wrists are revised again by (23). Finally, using inverse calculation (from endpoint to the former two joints, loop 2), we amend the shoulder and elbow data under the distance constraints (24).

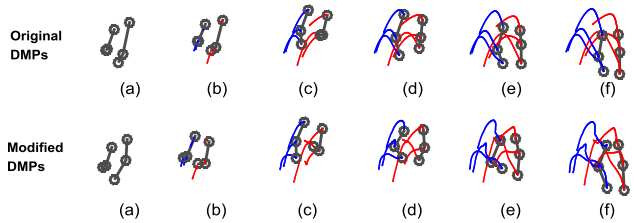



Fig. 8. Positions and trajectories of skeleton points in cooperative DMPs corresponding with the operation process (see (a)~(f)). The blue lines are generated skills of the left shoulder, elbow and wrist. The red lines are the results of the right arm.

The joint distance in fig.5(a) are measured with $\left\|L_{se}\right\|_2 = 22cm,$, $\left\|L_{ew}\right\|_2 = 28cm$, and $\left\|d_j^w\right\|_2 = 33cm$ is the length of the box and $h_j^s = h_j^e = h_j^w = 1cm, j = l,r$ are acceptable trajectory tracking errors and object deformation degree. The simulation results are presented in Figs. 8 to 10.

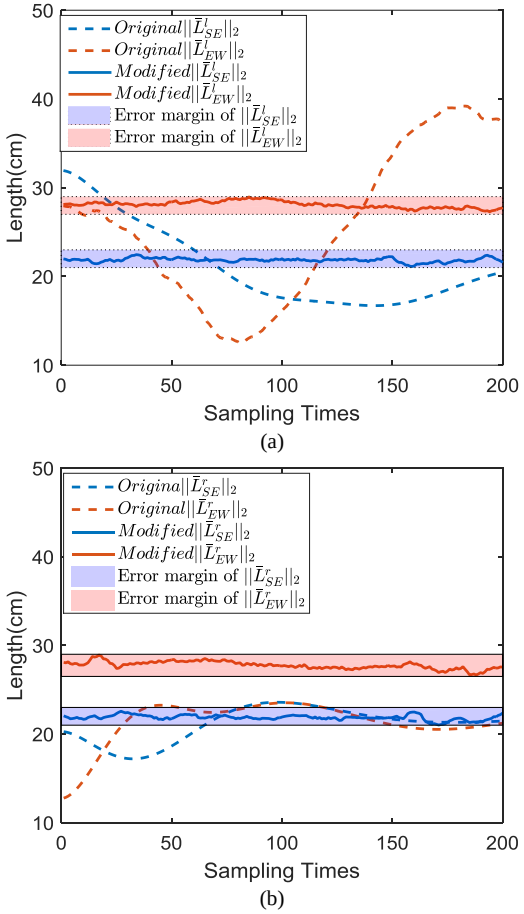



Fig. 9. Distance between adjacent joints (shoulder and elbow, elbow and wrist) (**a**) Results of the left arm (**b**) Results of the right arm. The solid lines of the two figures are the results of improved DMPs and the dash lines are the results of original DMPs with the same dataset, and the colored areas are designed error margins of the generated link lengths. The generated distance within the area means the constrained condition (21) is satisfied

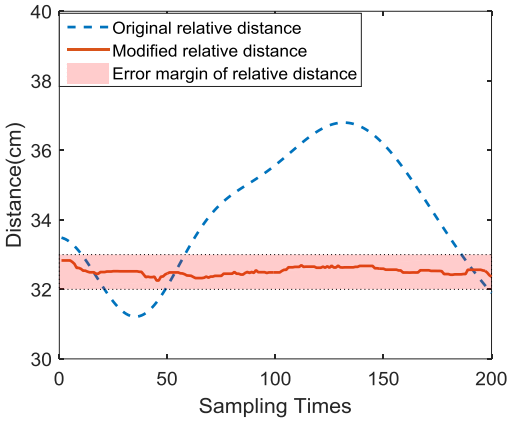



Fig. 10. Relative distance of the left hand and right hand in the generated DMPs path (red line is results of modified DMPs and blue line is results of original DMPs). The red area is the error margin of relative distance introduced in (19).

Fig. 8 shows the trajectories generated by constrained DMPs are different from original DMPs and data analysis is presented in Figs. 9 and 10. Fig.9 (a) shows the distance between adjacent joints of the left arm. It can be seen that the real joint distance of $\left\|\bar{L}_{se}^{l}\right\|_2$ and $\left\|\bar{L}_{ew}^{l}\right\|_2$ calculated by the original DMPs vary within ranges of $[12,40]$ and $[16,32]$, which are influenced seriously by the measuring noises and errors and can't meet the actual situation. So as the results of $\left\|\bar{L}_{se}^{r}\right\|_2$ and $\left\|\bar{L}_{ew}^{r}\right\|_2$ of the right arm in Fig.9 (b). While the trajectories of the modified DMPs keep a stable distance as predesigned in (24), which presents that the influence of noise is reduced. Fig.10 shows the relative distance of the wrist (hands). The modified DMPs framework decreases the varying range from $[31,37]$ to $[32,33]$, ensuring bimanual manipulation stability.

## VI. Conclusion

In this paper, we propose a general DMPs framework with the classified constraints based on a generalized DMPs model and BLFs. Compared with other improved DMPs models, a general additional term is calculated with certain expressions to encounter the changes of the environment and suit various tasks such as obstacle avoidance and cooperative manipulation by selecting suitable parameters. We make a further discussion about advantages of this method such as addressing dead-zone & human online preferred route, non-point obstacle and part of the measuring noise, as well as limitations. Finally, three experiments with some typical constraints are taken to verify the effectiveness and flexibility of the proposed method. The future work will combine the proposed method with auto navigation by using visual reality and the constraints will be learned and generated naturally from human demonstrations, not by artificial settings.

## Acknowledgment

The authors would like to thank Professor Qinchuan Li of Zhejiang Sci-Tech University (ZSTU) for his technical advice.

## References

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," *Handbook of robotics, Chapter 59, Springer*, 2008.

[2] A. K. Tanwani, and S. Calinon, "Learning robot manipulation tasks with task-parameterized semitied hidden semi-markov model," *IEEE Robot. Auto. Letters*, vol. 1, no. 1, pp. 235-242, 2016.

[3] S. M.Khansari-Zadeh, and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Trans. Robotics*, vol.27, no.5, pp. 943-957, 2011.

[4] S. Calinon, E. Sauser, Eric, D. Caldwell, and A. Billard, "A probabilistic approach based on dynamical systems to learn and reproduce gestures by imitation," *IEEE Robot. Autom. Mag.*, vol. 7, no.2, pp.44-54, 2009.

[5] A. Y. Ng, and S. J. Russell, "Algorithms for inverse reinforcement learning". in *Proc. Int. Conf. Machine Learn*., 2000, pp. 663–670.

[6] A. J. Ijspeert, Nakanishi J., and Schaal S., "Trajectory formation for imitation with nonlinear dynamical systems," in: *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2001. pp.752–757

[7] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no.2, pp.328-373. 2013.

[8] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Trans. Robot.*, vol. 26, no. 5, pp.800-815, 2010.

[9] Z. Li, T. Zhao, F. Chen, Y. Hu, C. Y. Su, and T. Fukuda, Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoid like mobile manipulator. *IEEE/ASME Trans. Mech.*, vol. 23, no. 1, pp.121-131, 2017.

[10] A. Colomé, and C. Torras, "Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes", *IEEE Trans. Robot.*, vol.34, no.3, pp.602-615, 2018.

[11] S. Calinon, and A. Billard, "Statistical learning by imitation of competing constraints in joint space and task space," *Adv. Robot.* vol.23, no. 15, pp. 2059-2076, 2009.

[12] L. Rozo, S. Calinon, and D. G. Caldwell, "Learning force and position constraints in human-robot cooperative transportation," in *23rd IEEE Int. Symp. Robot Human Interactive Commu.*, Aug. 2014, pp. 619-624.

[13] C. Yang, C. Chen, N. Wang, Z. Ju, J. Fu, and M. Wang, "Biologically inspired motion modeling and neural control for robot learning from demonstrations," *IEEE Trans. Cogn. Dev. Syst.*, vol. 11, no. 2, 2019, pp. 281-291.

[14] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives, " *IEEE Trans. Neur. Net. Learn. Syst.*, vol. 30, no. 3, 2019, pp. 777-787.

[15] S. M. Khansari-Zadeh, and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Auton. Robots*, vol.32, no.4, pp.433- 454, 2012.

[16] D. H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Humanoids 2008-8th IEEE-RAS Int. Conf. Humanoid Robots*. Dec. 2008, pp. 91-98.

[17] H. Hoffmann, P. Pastor, D. H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *2009 IEEE Proc. ICRA*, May, 2009, pp. 2587-2592.

[18] È. Pairet, P. Ardón, M. Mistry, and Y. Petillot, "Learning generalizable coupling terms for obstacle avoidance via low-dimensional geometric descriptors," *IEEE Robot. Auto. Letters*, vol.4, no.4, pp.3979-3986, 2019.

[19] A. Rai, F. Meier, A. Ijspeert, and S. Schaal, "Learning coupling terms for obstacle avoidance," in *2014 IEEE-RAS Int. Conf. Humanoid Robots*, Nov., 2014, pp. 512-518.

[20] J. Umlauft, D. Sieber and S. Hirche "Dynamic movement primitives for cooperative manipulation and synchronized motions," in 2014 *IEEE ICRA*, May. 2014, pp. 766-771.

[21] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Trans. Robot.*, vol.32, no.3, pp.513- 527. 2016.

[22] T. Kulvicius, M. Biehl, M.J. Aein, M. Tamosiunaite, and F. Wörgötter, "Interaction learning for dynamic movement primitives used in

cooperative robotic tasks," *Robot. Auton. Syst*., vol.61, no.12, pp.1450-1459, 2013.

[23] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Trans. Robot.*, vol. 30 no. 4, pp.816-830, 2014.

[24] G. Solak and L. Jamone, "Learning by demonstration and robust control of dexterous in-hand robotic manipulation skills," in *IEEE-IROS*, Macau, China, 2019, pp. 8246-8251.

[25] H. Lee, H. Kim, and H. J. Kim, "Planning and control for collision-free cooperative aerial transportation," *IEEE Trans. Autom. Sci. Eng*., vol. 15, no.1, pp.189-201, 2016.

[26] H. Qiao, M. Wang, J. H. Su, S. X. Jia, and R. Li, "The concept of 'attractive region in environment' and its application in high-precision tasks with low-precision systems," *IEEE/ASME Trans. Mech*., vol. 20, no. 5, pp. 2311-2327, 2015.

[27] L. Han, P. Kang, Y. Chen, W. Xu, and B. Li, "Trajectory optimization and force control with modified dynamic movement primitives under curved surface constraints." in *2019 IEEE-ROBIO*, 2019. pp. 1065-1070

[28] R. Krug, and D. Dimitrov, "Model predictive motion control based on generalized dynamical movement primitives," *J. Intell. Robot. Syst*., vol. 77, no.1, pp.17-35, 2015.

[29] C. Yang, X. Wang, L. Cheng and H. Ma, "Neural-learning based telerobot control with guaranteed performance, " *IEEE Trans. Cyber.*, vol. 47, no. 10, pp.3148-3159, 2017.

[30] G. Peng, C. Yang, W. He and C. L. P. Chen, "Force sensorless admittance control with neural learning for robots with actuator saturation", *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3138-3148, 2020.

[31] H. Huang. T. Zhang, C. Yang and C. L.P. Chen, "Motor Learning and Generalization Using Broad Learning Adaptive Neural Control", *IEEE Trans. Ind. Electron.* vol. 67, no. 10, pp. 8608-8617, 2020.

[32] R. Wang, Y. Wu, W. L. Chan, and K. P. Tee, "Dynamic movement primitives plus: For enhanced reproduction quality and efficient trajectory modification using truncated kernels and local biases, " *In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2016, pp. 3765-3771.

[33] Y. Wu, R. Wang, L. F. D'Haro, R. E. Banchs, and K. P. Tee, "Multi-modal robot apprenticeship: Imitation learning using linearly decayed dmp+ in a human-robot dialogue system." *In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 8582-8588.

[34] A. Duan, R. Camoriano, D. Ferigo, D. Calandriello, L. Rosasco, and D. Pucci, Constrained DMPs for feasible skill learning on humanoid robots. *In 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* Nov. 2018, pp. 1032-1038.

[35] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Trans. Cyber*., vol. 46, no.3, 2015. pp.620-629.

[36] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp.918-927, 2009.

[37] A. Elnagar, and K. Gupta, "Motion prediction of moving objects based on autoregressive model," *IEEE Trans. Syst., Man, Cyber. -Part A: Syst. Humans*, vol. 28, no. 6, pp. 803- 810, 1998.

[38] H. Huang, C. Yang, and C. L. P. Chen, "Optimal robot-environment interaction under broad fuzzy neural adaptive control, " In press. *IEEE Transactions on Cybernetics*. DOI:10.1109/TCYB.2020.2998984.

[39] J. Fu, T. Chai, C. Su, and Y. Jin, "Motion/force tracking control of nonholonomic mechanical systems via combining cascaded design and backstepping, " *Automatica*, vol. 49, no.12, pp. 3282-3286, 2013.

[40] M. Ogrinc, A. Gams, T. Petrič, N. Sugimoto, A. Ude, and J. Morimoto, "Motion capture and reinforcement learning of dynamically stable humanoid movement primitives," in *IEEE ICRA*, pp. 5284-5290, 2013.

[41] R. Elbasiony, and G. Walid "Humanoids skill learning based on real-time human motion imitation using Kinect," *Intell. Serv. Robot.*, vol. 11, no. 2, pp.149-169, 2018.

[42] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, "A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills," *IEEE/ASME Trans. Mech*., vol. 23, no.3, pp. 1193-1203, 2018.

[43] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, "A learning framework of adaptive manipulative skills from human to robot," *IEEE Trans. Ind. Inform.*, vol.15, no. 2, pp. 1153-1161, 2018.

[44] C. Zeng, C. Yang, J. Zhong, and J. Zhang, "Encoding multiple sensor data for robotic learning skills from multimodal demonstration, " *IEEE Access*, vol. 7, pp. 145604-145613, 2019.